REMARKS

Claims 1 and 3 have been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,324,565 to Holt III ("Holt"), in view of U.S. Patent Publication No.: 2002/0004813 to Agrawal et al. ("Agrawal") and further in view of U.S. Patent Application Publication No. 2004/0064570 to Tock ("Tock").

Claims 2 and 4 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Holt in view of Agrawal and Tock, and further in view of U.S. Patent No. 6,457,047 to Chandra et al. ("Chandra").

Claims 1-4 remain pending.

Rejection of Claims 1 and 3 under 35 U.S.C. Section 103(a)

With respect to independent claim 1, the Office Action states that Holt teaches substantially the invention as claimed except caching XML elements in a document object model tree according to a cache strategy including at least one of depth, medium, weight, and scale; and packaging XML elements into an XML document, which Agrawal allegedly teaches. The Office Action also states that Holt does not teach that the client can be one of a plurality of types of devices, which Tock allegedly teaches.

Holt discloses a system for dynamically generating documents that utilizes document programs and data at a content providing server to generate those documents. When an intermediate server requests a dynamically generated document from a content providing server, the content providing server transmits the instructions or programs to create the document along with data utilized by the program or instructions in creating the document. The intermediate server then retains or caches the document programs as well as the data. When the intermediate

server next requires the document, that document is generated at the intermediate server rather than requiring that it be obtained from the content providing server. The content providing server retains a register of the intermediate servers, which have received document programs or data. If there are changes to the document programs or data, such changes are broadcast to the intermediate servers, which had cached that information.

Holt teaches only retrieving information to generate documents and returning them to a client. Holt does not teach or suggest specifically retrieving information and generating documents in XML format to produce objects that are compatible with, and retrievable by, a plurality of types of devices. Holt is specifically limited to retrieving documents for client computers. Holt also does not teach caching XML elements in a document object model tree according to a cache strategy including at least one of depth, medium, weight, and scale.

Agrawal discloses a method of servicing a request for a document over a computer network. The method includes independently caching portions of pages called blocks. Each block includes a reference to a data source and code that is adapted to access the data source and to format the data accessed from the data source. When a request for a page is received over a computer network, one or more of the plurality of blocks defined in the script of the requested document may be retrieved from a cache memory. Any block that is not found in the cache memory is dynamically generated and a copy thereof is stored in the cache memory. The requested page may then be assembled from the page blocks retrieved from the cache memory and/or the dynamically generated page blocks.

The Examiner cites paragraphs 28, 29, and 34 of Agrawal as teaching wherein XML elements are in a document object model tree according to a cache strategy including at least one

of depth, medium, weight, and scale. Applicants submit that the cited passages have been misinterpreted.

Paragraph 28 of Agrawal discloses a Page Block or Block. A document, such as an XML or HTML page, when rendered in a browser on a computer monitor, may be interpreted as a collection of blocks. In a Document Object Model of the page, for example, such blocks refer to different nodes of the DOM tree. Page blocks may also be considered from a higher level. For example, a Web page that contains both stock quotes and an email notification section may be defined in terms of a stock quote block and an email notification block. A Web page, therefore, may be defined in terms of its constituent page blocks. Paragraph 28 only discloses using a document object model tree. Nothing is taught or suggested regarding a caching strategy including at least one of depth, medium, weight, and scale.

Paragraph 29 of Agrawal discloses caching of HTTP responses at a different level of granularity than full page caching. The granularity of caching is a "Page Block" instead of a full page. The pages from which blocks are cached may be, or include, any XML document. Paragraph 29 only discloses caching in units smaller than a full page. Nothing is taught or suggested regarding a caching strategy including at least one of depth, medium, weight, and scale.

Paragraph 34 of Agrawal discloses that each page block may be assigned or have caching properties associated therewith. The caching properties determine what the block retrieval key for the cached block should be, the invalidation mechanism, the expirations times and the like. For example, the block retrieval key may include a unique identifier that uniquely identifies each cached block from among all other cached blocks. Since the granularity of the cache is at the block level, the disclosed caching mechanism is able to handle such situations. In case of

conventional full page caching, the expiration time for the full page is the minimum of the expiration times for each of the data components thereof. Therefore, if one of the data components of a Web page is changing very frequently or non-deterministically, the full page cache will not be able to efficiently cache the page, as it would be invalidated very quickly, as the data contained therein stales. In contrast, only those blocks whose data have changed will be invalidated, while the still-valid cached blocks remain unaffected and fully able to be retrieved and used in assembling a page responsive to an HTTP request. The system disclosed in Agrawal provides for retrieving the cached constituent blocks thereof that are still valid from the cache memory and dynamically generating any block in the script of the requested page that is not stored in the cache memory. A copy of each dynamically generated block may then be stored in the cache memory. The requested page may then be assembled from the retrieved block(s) retrieved from the cache memory and/or the dynamically generated blocks. This assembly may take place at the server that services the user's HTTP request or the constituent blocks of the requested page may be sent over the computer network to be assembled by the client browser. Paragraph 34 only discloses invalidating blocks, and storing and retrieving blocks from cache. Nothing is taught or suggested regarding a caching strategy including at least one of depth, medium, weight, and scale.

In contrast, Applicants' claim 1 recites a cache for caching the XML elements in a document object model tree according to a cache strategy including at least one of depth, medium, weight, and scale.

Performing caching according to the depth involves caching only elements of low level (for example the first and second depth) to save memory. Performing caching according to medium involves caching against the medium that requires a minimum of memory capacity (for

example texts or black and white images). Performing caching according to weight involves setting different weight values for elements of different depths or different medium. For weight caching, caching is only performed against elements with weighted values lower than a preset value. Performing caching according to scale involves caching elements with different weighted values in different caches of different levels (See page 11, line 20 to page 12, line 10 of the originally filed specification). Nothing in Agrawal teaches or suggests such caching strategies.

Tock teaches a system and method for enabling a client application to operate offline from the server. For each request, the client application registers a request entry with a scheduler, informing the scheduler of an associated application callback. The scheduler then invokes this application callback when a network connection is available for use, and the client application then sends the request to the server. Error recovery methods are described for requests that are sent to the server but for which no reply is received, and for other error scenarios.

Further, the passages cited by the Examiner on page 8, paragraph 85 refer to a web browser or client side application being capable of executing on any type of client computer. The cited passages do not teach or suggest retrieving XML elements and packaging the XML elements into a XML document and sending it back to the at least one of the plurality of types of devices. Moreover, Tock also does not teach or suggest caching XML elements in a document object model tree according to a cache strategy including at least one of depth, medium, weight, and scale.

In contrast, Applicants' invention teaches a system and apparatus for providing instant information service for a plurality of types of devices. The invention includes a network connecting unit that fetches data from backend servers and packages the data into XML

elements. The invention further includes a controller and a cache, which caches the XML elements formed by the network connecting unit. The XML elements are cached in a document object model tree according to a cache strategy including at least one of depth, medium, weight, and scale. The controller fetches relevant XML elements from the cache in response to a request for information service from at least one of the plurality of types of devices. When elements cannot be fetched from the cache, the controller also instructs the network connecting unit to fetch corresponding data from backend servers and obtains the XML elements formed by the network connecting unit. Further, the controller packages all the fetched XML elements into an XML document and sends it back to the at least one of the plurality of types of devices. XML is chosen as the format for the elements because XML is compatible with a plurality of device types, such as computers, cell phones, and wired telephones, for example

Further still, there is nothing taught or suggested in Holt, Agrawal, or Tock that creates a motivation to combine the references. The Examiner cannot base obviousness upon what a person skilled in the art could, or might, try but rather must consider what the prior art would have led a person skilled in the art to do. In re Antonie, 559 F.2d 618 195 USPQ 6 (CCPA, 1977). To prevent the use of hindsight based on the invention to defeat patentability of the invention, the Examiner must show a motivation to combine the references that create the case of obviousness. In re Rouffet, 47 USPQ2d 1453 (Fed. Cir., July 15, 1998). The conclusion asserted by the Examiner represents an impermissible use of hindsight gained from the present invention.

In view of the foregoing, it is respectfully submitted that Holt, Agrawal, and Tock, whether taken alone or in combination, do not teach or suggest the subject matter recited in claim 1 as each of these references fails at least to teach or suggest an apparatus for providing instant

information service for a plurality of types of devices, where the apparatus includes a cache for caching the XML elements in a document object model tree according to a cache strategy including at least one of depth, medium, weight, and scale, the XML elements formed by the network connecting unit by packaging, and where data is retrieved and packaged into XML elements that are compatible with, and retrievable by, a plurality of types of devices.

Independent claim 3 recites similar features as claim 1, and therefore is patentably distinct over Holt, Agrawal, and Tock for at least the reasons discussed in connection with claim 1.

Rejection of Claims 2 and 4 under 35 U.S.C. Section 103(a)

The Office Action states that Holt teaches an invention where documents for clients are generated from data obtained from the cache and the backend server to reduce network traffic. However, Holt does not teach of an indexing mechanism for creating indices for all the XML elements stored in the cache. The Office action further states that Chandra teaches a centrally maintained table in the cache directory for determining if the query is cached.

Holt, Agrawal, and Tock have been previously described and, whether taken alone or in combination, do not teach or suggest the subject matter recited in Applicants' independent claims 1 and 3. Further, because Holt, Agrawal, and Tock do not teach or suggest the subject matter recited in independent claims 1 and 3, and because Chandra does not teach or suggest the elements of claims 1 and 3 that Holt, Agrawal, and Tock are missing, Chandra is irrelevant.

Claims 2 and 4, which depend directly or indirectly from the independent claims 1 and 3 incorporate all of the limitations of the corresponding independent claim and are therefore

patentably distinct over Holt, Agrawal, Tock, and Chandra for at least those reasons provided for claims 1 and 3.

Conclusion

While consideration of this response is not a matter of right, the Examiner, in his rejection of all claims in the Final Rejection dated January 9, 2006 cited, cited the new reference to Agrawal. Applicants thus respectfully request entry of this response and submit that the citation of the new reference constitutes good and sufficient reason why the foregoing remarks could not have been earlier presented.

In view of the foregoing, Applicants respectfully requests reconsideration, withdrawal of all rejections, and allowance of all pending claims in due course.

Respectfully submitted,

Steven Fischman
Registration No. 34,594

Scully, Scott, Murphy & Presser
400 Garden City Plaza, Suite 300
Garden City, N.Y. 11530
(516) 742-4343

SF/BMM:ej

G:\IBM\105\14657\amend\14657.am4.doc